The Pixels Array

3 min

Before diving into more complex image manipulation techniques, let's look at what's happening under the hood with the `get()` and `set()` functions.

The `pixels` array is a special data structure that enables us to access and manipulate pixel data—whether in the canvas or in individual images or videos. The `get()` and `set()` functions are actually functions that make working with pixels array easier, but slower.

We normally think of an image as a two-dimensional matrix—an array of arrays—with each pixel having a red, green, blue, and alpha (RGBA) value.

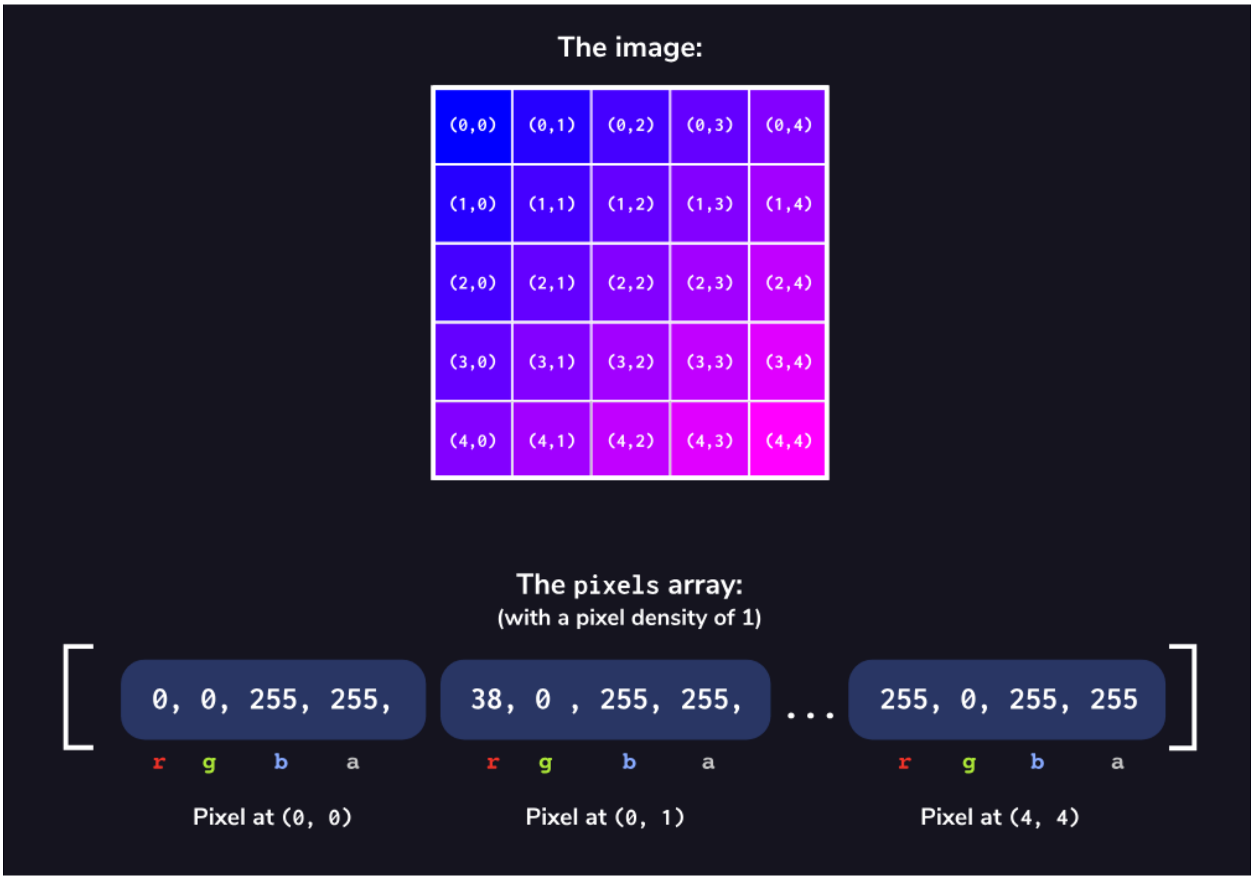
The `pixels` array is a representation of all those pixels, flattened down to a single list. Starting from the first pixel and moving left-to-right by rows to the last pixel, `pixels` stores the RGBA values for each pixel into one, single array. The one-dimensional array format, while harder to work with, makes accessing the data fast.

The pixels array is also more like a snapshot of a canvas or image, rather than a source of truth for what's on the screen. This is where the `updatePixels()` function comes back in—after we modify the `pixels` array, it's how we indicate that we're done making changes, and we're ready to see it updated on the sketch.

In some cases, the `pixels` array will be affected by *pixel density*. The pixel density in a p5.js sketch depends on your computer monitor—many high-resolution monitors treat a single pixel on the screen as multiple smaller pixels, creating a sharper image—accordingly, the number of elements in the `pixels` array factors in that density.

Instructions

View the diagram to the right to see how the `pixels` array consists of the RGBA values of each pixel in an image.



The image:

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)
(4,0)	(4,1)	(4,2)	(4,3)	(4,4)

The `pixels` array:
(with a pixel density of 1)

```
[ 0, 0, 255, 255,   38, 0 , 255, 255,  ...  255, 0, 255, 255 ]
  r  g  b    a      r   g   b    a          r    g  b    a
  Pixel at (0, 0)   Pixel at (0, 1)         Pixel at (4, 4)
```